

Migration of LArSoft repos to GitHub

Patrick Gartung
LArSoft Coordination Meeting
10 Sept 2019

Motivation

- Move repos to Github for easier collaboration
- Move to the Github development model (see next slide)
- Take advantage of the resources of Github
- Use Github's web based tools: code reviews, etc.

Github development model

- User creates primary repos on Github under your user account or organization.
- Anyone can fork your primary repos and make changes in their forked repo.
- Changes are proposed through pull requests from the fork to the primary.
- Primary repo can be configured to trigger CI, etc., on pull request creation
- Pull request can be reviewed, revised, approved, merged, etc through the web interface.
- Pull request can be merged by users with the appropriate permissions.

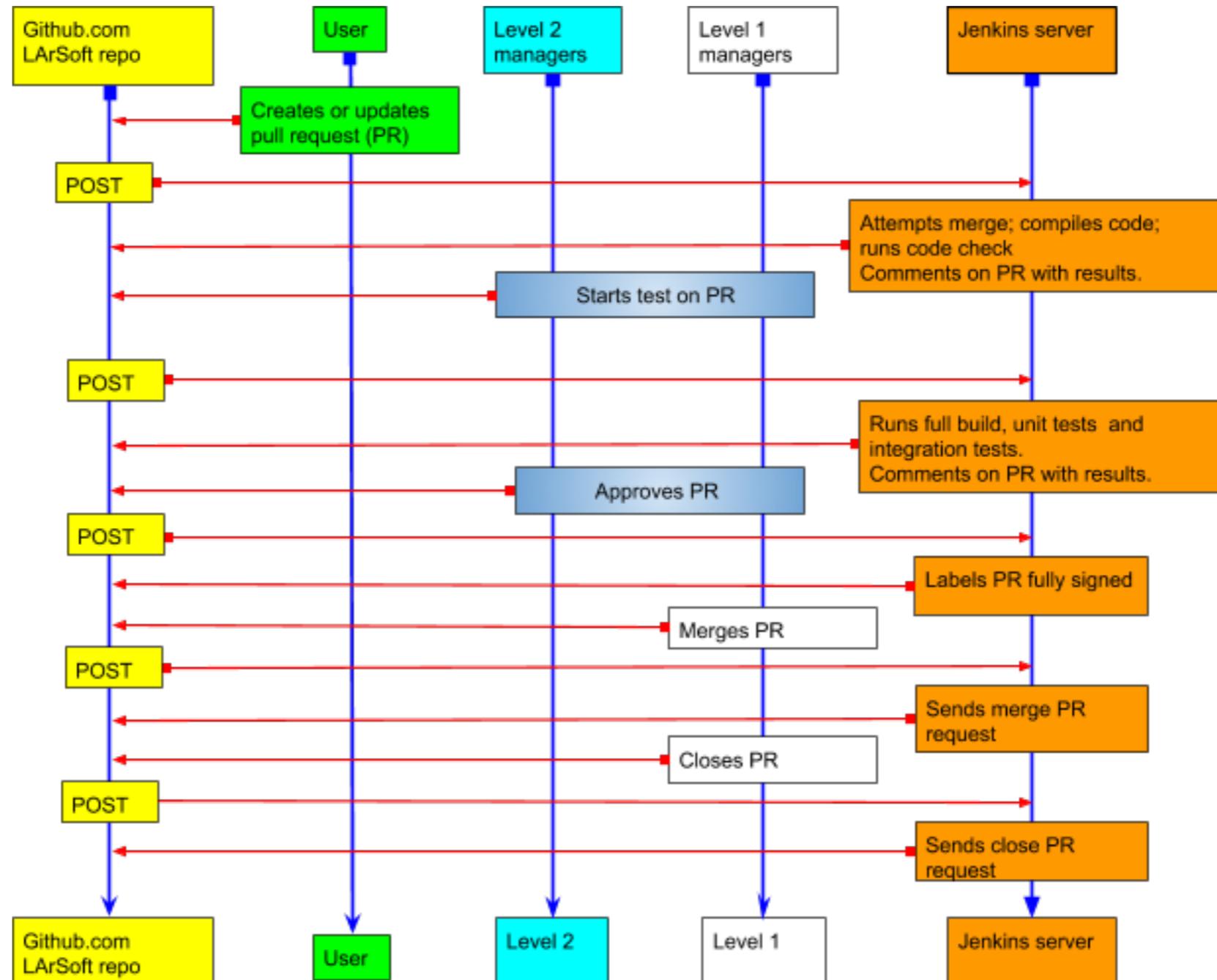
Migration plan

- Investigate the CMS system of pull request testing, approval and integration.
- Re-use as much of the CMS scripts as possible to achieve this for LArSoft repos.
- Cleanup LArSoft repos' history in a reproducible manner.
- Create an organization of Github for the LArSoft repos.
- Migrate the LArSoft repos to Github.
- Make Redmine repos read only.
- Start using Github repos for source checkout.

Re-using CMS functionality

- CMS made the migration from CVS directly to git and Github over 5 years ago and uses Github messaging to manage pull requests with a suite of scripts.
- Scripts are tailored to how CMSSW is built: one monolithic code repo vs many code repos
- Python scripts using py-github to post messages to a pull request “Conversation” with the results of pull request CI tests and labels to indicate the status of a pull request.
- Defines two lists of “managers”:
 - “Level 2” managers can trigger pull request tests and must sign off on the pull request
 - “Level 1” managers can merge the pull request to the default branch
- Per policy, pull requests must be approved by managers before they are merged.
- A small subset of these scripts were reusable, with modifications, for use with LArSoft repos.
- The commands users can use in messages is described on this web page
 - <http://cms-sw.github.io/cms-bot-cmssw-cmds.html>
- The pull request messaging sequence implemented by these scripts is shown on the next slide.

Pull request messaging sequence



- Github notifies anyone mentioned with @username in a comment by email when comments are added to the “Conversation”
POST = send a message with data to jenkins

Infrastructure

- Requested a publicly accessible website to host the CGI script to accept POST messages from Github and trigger a Jenkins job if the signature passes verification and the author is a manager. https://scd-ci.fnal.gov/cgi-bin/github_webhook
- Set up a Jenkins job, dispatch-github-webhook, to act on data from Github by triggering other Jenkins jobs.
- Requested that FNAL Jenkins server allow connections from Github. This was granted and built-in Jenkins Github plugin can be used. However, they do not reproduce the signature features of the CMS scripts.
- Created a [LArSoft organization on Github](#).
- Created [Core group in LArSoft organization](#) which is assigned admin privileges for all repos.
- Set up Github web hooks for all repos so messages would be sent when pull requests are created. This is the equivalent of a git post commit script.
- Created a Github account [FNALbuild](#) and generated access tokens. Jenkins jobs can post messages to a pull request “Conversation” as this account using the access tokens.
- Created a mailing list scd-github-fnalbuild@listserv.fnal.gov and set this as email for the FNALbuild account. Email is sent to the list when a new PR is created for any repo the account is “Watching”.

Repo cleanup needed

- Github restricts the size of any file in a commit to less than 100MB.
- All of the original LArSoft repos have an initial commit with a ~105MB xml file in the initial commit. The xml file is highly compressible but Github looks at the uncompressed size.
- Truncating the git history at the svn to git transition removes the large file from the commit history. Removing large root and log files from the LArSim repo completes the cleanup.

Repo Size Reduction

| repo | before | after | before | after |
|-----------------|--------|-------|---------|---------|
| | size | size | commits | commits |
| larana | 18M | 17M | 6521 | 1266 |
| larcore | 27M | 27M | 6032 | 777 |
| larcorealg | 29M | 18M | 903 | 732 |
| larcoreobj | 2.3M | 2.2M | 163 | 163 |
| lardata | 19M | 18M | 6893 | 1638 |
| lardataalg | 99M | 99M | 416 | 416 |
| lardataobj | 1.3M | 1.2M | 545 | 543 |
| lareventdisplay | 17M | 16M | 6246 | 990 |
| larevt | 17M | 16M | 5930 | 674 |
| larexamples | 16M | 15M | 5786 | 530 |
| larg4 | 0.4M | 0.4M | 151 | 150 |
| larpandora | 19M | 18M | 6177 | 919 |
| lareco | 31M | 30M | 11899 | 6629 |
| larsim | 2000M | 86M | 7027 | 1770 |
| larsoft | 19M | 18M | 6521 | 1263 |
| larsoftobj | 0.6M | 0.5M | 297 | 297 |
| larwirecell | 0.8M | 0.7M | 364 | 363 |

Source checkout with Github repos

- Development checkout needs
 - Determine a users GitHub account, get an access token and save both for use at another time.
 - Fork a primary LArSoft repo to a users GitHub account and set their fork as the origin with the primary as upstream.
- CMS solved this with git extensions, git-cms-checkout, etc.
- Found a utility, called hub, that performs these functions.
- Proposed change to MRB gitCheckout script to add support to checkout LArSoft sources from GitHub using the hub utility to manage the forking of the LArSoft primary repos

Conclusions

- Can reproduce the pull request messaging used by CMS to trigger CI tests when pull requests are created
- The infrastructure on Github has been created.
- The repo cleanup needed before migrating the LArSoft repos to Github is understood and reproducible.
- A utility to handle interaction with GitHub from the command line has been identified.
- Many technical issues with coordinating pull requests from multiple repos still need to be resolved.
- This is an ongoing effort. A detail deployment plan and documentation is in the works.